SecureForge Web: Avaliação Guiada e Hardening de Aplicações Web com Análise Assistida

Margefson M. Barros [1], Nattan S. Lobato [1], Josias S. Bentes[1], Keven Mendes[1], Eduardo Souto [1], Eduardo L. Feitosa[1]

[1]Instituto de Computação – (ICOMP) – Universidade Federal do Amazonas (UFAM)
CEP – 69.077-000 – Manaus – AM – Brasil

{margefson.barros,nattan.lobato,josias.bentes,kevencoimbra

esouto,efeitosa}@icomp.ufam.edu.br

Modalidade: Código Aberto

Abstract. *SecureForge Web is a web platform for guided security assessment and gradual hardening of web applications. Unlike professional vulnerability scanners, it combines an OWASP/ASVS-aligned checklist, human-in-the-loop validation, and assisted automation through HTTP header inspection, static repository analysis, and a per-user AI assistant. The tool records structured evidence per checklist item, generates findings with remediation guidance, tracks security posture over time, and exports PDF reports. We describe its architecture, assessment modules, and an experimental evaluation on twelve web applications — intentionally vulnerable labs, open-source demos, and the platform itself — demonstrating end-to-end operation from registration to posture benchmarking.*

Resumo. *A SecureForge Web é uma plataforma web para diagnóstico guiado e fortalecimento gradual (*hardening*) de aplicações web. Diferentemente de scanners profissionais de vulnerabilidades, combina checklist alinhado a OWASP/ASVS, validação humana no fluxo e automação assistida por inspeção de headers HTTP, análise estática de repositório Git e assistente de IA configurável por usuário. A ferramenta registra evidências estruturadas por item de checklist, gera achados com recomendações de correção, acompanha a postura de segurança ao longo do tempo e exporta relatórios em PDF. Descrevemos sua arquitetura, módulos de avaliação e uma avaliação experimental sobre doze aplicações web — laboratórios intencionalmente vulneráveis, demos de software livre e a própria SecureForge Web — demonstrando operação ponta a ponta desde o cadastro até o* benchmark *de postura.*

1. Introdução

A avaliação da segurança de aplicações web envolve controles relacionados a autenticação, autorização, gestão de sessões, validação de entrada, proteção de dados e configuração da aplicação. Padrões como *Application Security Verification Standard* (ASVS) [OWASP 2025] e *Web Security Testing Guide* (WSTG) [OWASP 2021], ambos da OWASP, organizam requisitos e procedimentos para essa avaliação. Entretanto, transformar essas recomendações em um processo operacional, repetível e rastreável ainda

pode ser difícil, especialmente para equipes sem processos consolidados de segurança de software.

Abordagens automatizadas têm sido propostas para identificar vulnerabilidades e sistematizar testes de segurança [Akrout et al. 2014, Zech et al. 2019]. Contudo, sua capacidade varia conforme o controle avaliado. Problemas relacionados a permissões, regras de negócio e comportamentos esperados da aplicação frequentemente exigem interpretação contextual e revisão humana [Rennhard et al. 2022]. Além disso, modelos baseados no ASVS podem consolidar os resultados em indicadores de postura, mas não eliminam a necessidade de coletar evidências, revisar classificações e gerenciar os problemas identificados [Wen and Katt 2023].

Este artigo apresenta a SecureForge Web, uma plataforma para avaliação guiada e *hardening* de aplicações web. A ferramenta combina perfis de *checklist*, inspeção de respostas HTTP, análise estática de repositórios Git e assistência por modelos de linguagem. Para cada item, os módulos da ferramenta podem sugerir uma classificação, justificativa e evidências técnicas, que são revisadas pelo analista antes do registro da decisão final. Itens parcialmente conformes ou não conformes podem gerar achados com severidade, prioridade e recomendações de correção.

As principais contribuições deste trabalho são: (i) um fluxo integrado de avaliação baseado em *checklist*, análise assistida e validação humana; (ii) mecanismos para geração e persistência de evidências associadas aos itens avaliados; e (iii) um estudo de caso com 12 alvos, incluindo aplicações intencionalmente vulneráveis, instâncias públicas de projetos de código aberto e duas configurações da própria SecureForge Web.

A contribuição deste trabalho é de *integração e operacionalização*, e não a proposta de um novo método de descoberta de vulnerabilidades. Scanners e proxies (p.ex. OWASP ZAP, Burp Suite, Probely) e analisadores estáticos (p.ex. Semgrep) cobrem exploração e regras de código; plataformas como DefectDojo, Faraday, Dradis e o Security Knowledge Framework (SKF) cobrem agregação de achados, pontuação ASVS ou conteúdo educativo. A SecureForge Web distingue-se por materializar um fluxo item a item com evidências estruturadas, preservação separada entre sugestão automática e decisão humana, e consolidação em postura/PDF, sem substituir aqueles motores.

2. SecureForge Web

A SecureForge Web organiza a avaliação de segurança de aplicações web como um fluxo guiado que se inicia com o cadastro da aplicação e a seleção de um perfil de *checklist*. O cadastro reúne a URL da aplicação, seu repositório de código-fonte, quando disponível, e a tecnologia utilizada. A plataforma oferece um checklist próprio e perfis baseados no OWASP ASVS. Neste trabalho, utiliza-se o *Checklist Essential SecureForge v1.0*, perfil operacional de 24 itens em nove categorias: Autenticação, Autorização, Validação de entrada, Proteção de credenciais, Headers de segurança, Exposição de endpoints, Mensagens de erro, Dados sensíveis e Superfície de ataque. Cada item possui um código interno (p.ex. `HEADER-01`) e é mapeado a um requisito canônico do ASVS 5.0.0 no formato `v5.0.0-X.Y.Z` (p.ex. `HEADER-01` → `v5.0.0-3.4.3`), preservando a rastreabilidade entre versões do padrão. O Essential não substitui os catálogos ASVS Level 1 e Complete, também importáveis na ferramenta; trata-se de um recorte para avaliação rápida por equipes de baixa maturidade em AppSec.

A Tabela 1 ilustra o mapeamento para itens representativos (O mapeamento completo dos 24 itens - código, categoria, enunciado e `v5.0.0-X.Y.Z` - está publicado no repositório da ferramenta[1]). Ela exemplifica a correspondência entre itens do Essential v1.0 e identificadores ASVS 5.0.0.

Tabela 1. Exemplos de rastreabilidade Essential v1.0 ↔ OWASP ASVS 5.0.0.

Código	ASVS 5.0	Item
AUTH-01	`v5.0.0-6.2.1`	Política de senha mínima
AUTHZ-01	`v5.0.0-8.2.1`	Controle de acesso por perfil
INPUT-02	`v5.0.0-1.2.4`	Queries SQL parametrizadas
SECRET-01	`v5.0.0-13.3.1`	Segredos em variáveis de ambiente
HEADER-01	`v5.0.0-3.4.3`	Content-Security-Policy
DATA-01	`v5.0.0-12.2.1`	HTTPS/TLS em trânsito

Durante a avaliação, módulos de análise assistida da ferramenta examinam respostas HTTP, o repositório Git e informações contextuais para produzir sugestões, justificativas e evidências. O analista revisa essas informações e registra a classificação final de cada item. Itens parcialmente conformes ou não conformes podem gerar achados com severidade, prioridade e recomendações de correção. Ao final, a plataforma consolida os resultados em indicadores de postura, apresentados no dashboard consolidado (Figura 1), e mantém o histórico necessário para avaliações posteriores.

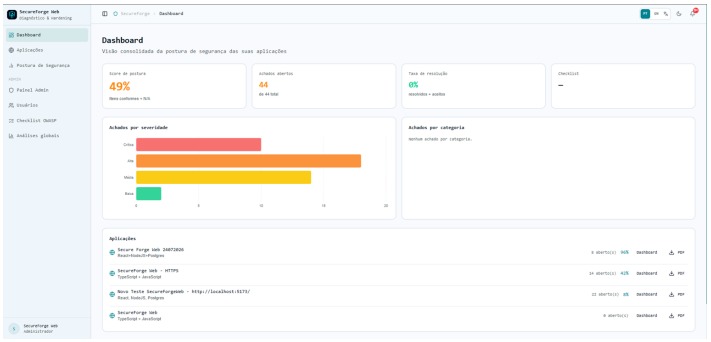

Figura 1. Dashboard consolidado: score de postura, achados e exportação PDF.

2.1. Visão Geral e Arquitetura

A arquitetura da SecureForge Web, apresentada na Figura 2, é organizada em quatro componentes: interface web, núcleo da aplicação, módulos de análise assistida e camada de persistência.

A interface web oferece o fluxo de cadastro, seleção do *checklist*, revisão dos itens e acompanhamento dos achados. O núcleo coordena as avaliações, gerencia usuários e aplicações e integra os módulos HTTP, Git e baseado em modelo de linguagem. Esses módulos consultam fontes distintas e produzem sugestões acompanhadas de justificativas e evidências.

A camada de persistência armazena aplicações, perfis de *checklist*, respostas, sugestões, evidências, decisões dos analistas e achados. Esse histórico preserva a rastreabilidade das classificações e permite comparar avaliações sucessivas da mesma aplicação.

[1] `https://github.com/secureforgeweb/secureforgeweb`

Como indicado na Figura 2, os módulos apenas apoiam a análise, a classificação final permanece sob responsabilidade do analista.

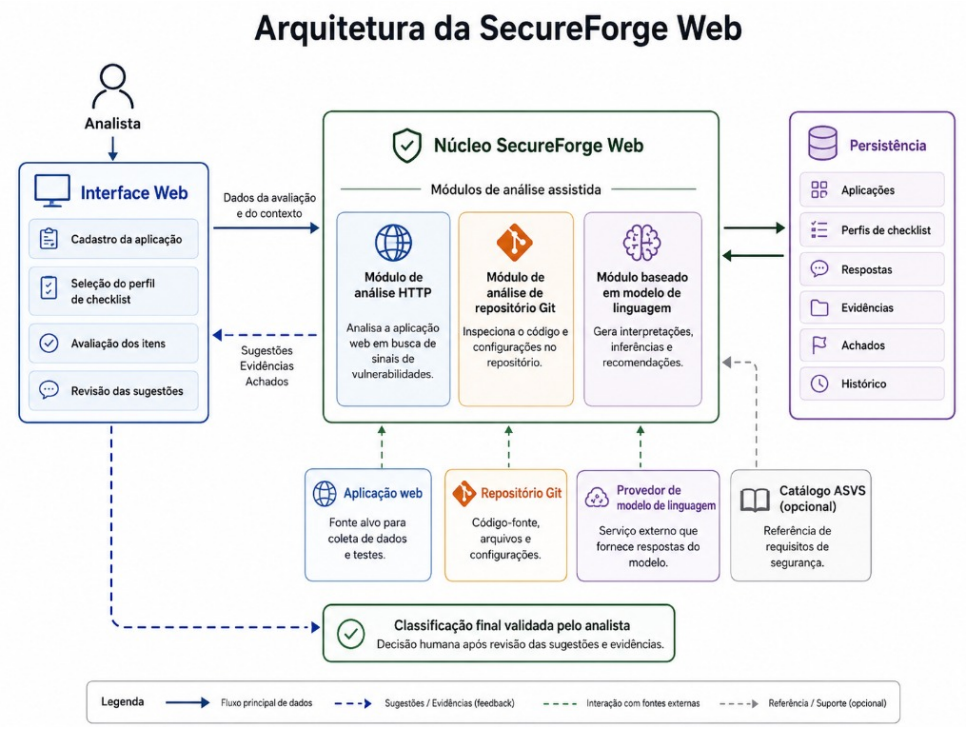

Figura 2. Arquitetura e fluxo de avaliação da SecureForge Web.

2.2. Módulos de Análise Assistida

A SecureForge Web dispõe de três módulos de análise assistida, que utilizam como fontes a resposta HTTP da aplicação, seu repositório de código-fonte e, opcionalmente, um modelo de linguagem. Os módulos produzem sugestões, justificativas e evidências para os itens do *checklist*, mas não substituem a decisão do analista.

O módulo HTTP realiza uma requisição à URL cadastrada e examina a resposta final após eventuais redirecionamentos. São verificados aspectos como uso de HTTPS e configuração de *Content-Security-Policy*, *Strict-Transport-Security*, proteção contra *clickjacking* e *X-Content-Type-Options*. Para cada item, o módulo registra a classificação sugerida, o nível de confiança, a justificativa e os cabeçalhos utilizados como evidência.

Antes de efetivar a requisição, a URL cadastrada e cada redirecionamento são validados: bloqueiam-se destinos em faixas privadas (RFC 1918), link-local e serviços de metadados de nuvem (p.ex. `169.254.169.254`), bem como hostnames de metadados; a resolução DNS é reavaliada para reduzir risco de *rebinding*. Endereços de *loopback* permanecem permitidos para laboratórios locais. Avaliações em LAN exigem configuração explícita (`ALLOW_PRIVATE_ASSESSMENT`).

O módulo Git realiza uma inspeção estática do repositório associado à aplicação. Heurísticas procuram padrões relacionados a políticas de senha, armazenamento de credenciais, limitação de requisições, expiração de sessões, controle de acesso, validação de entradas, consultas parametrizadas, proteção contra XSS (*Cross-Site Scripting*), gestão de

segredos e tratamento de erros. Quando identifica um padrão relevante, o módulo registra como evidência o arquivo, o intervalo de linhas e um trecho do código-fonte.

O módulo baseado em modelo de linguagem complementa as informações produzidas pelos demais módulos. Quando configurado, recebe o item avaliado e as evidências disponíveis e pode sugerir uma classificação, justificativa e recomendação em formato estruturado. Na ausência de configuração válida ou em caso de falha na comunicação com o provedor, a avaliação prossegue com os demais módulos.

Os módulos têm coberturas distintas e limites práticos: o HTTP restringe-se ao observável na resposta (sem controles internos/regra de negócio); o Git depende das linguagens e heurísticas; o LLM pode errar. Por isso a plataforma registra a origem de cada sugestão e exige revisão humana. No wizard, as ações *Analisar repositório Git*, *Assistente IA* e análise HTTP são independentes por categoria; a classificação final permanece com o analista.

2.3. Evidências, Validação Humana e Gestão de Achados

Para cada item, a SecureForge Web armazena a sugestão produzida, sua origem, o nível de confiança, a justificativa e as evidências disponíveis. Essas evidências podem incluir cabeçalhos HTTP, URL e protocolo observados, caminhos e trechos de arquivos, intervalos de linhas e informações geradas pelo modelo de linguagem.

Durante a avaliação, o analista examina essas informações e registra a classificação final, podendo aceitar, modificar ou complementar a sugestão. A plataforma preserva separadamente o resultado dos módulos e a decisão humana, mantendo a rastreabilidade do processo. A Figura 3 apresenta um exemplo dessa interface.

Itens parcialmente conformes ou não conformes podem gerar achados associados ao item de origem. Cada achado reúne severidade, prioridade, recomendação de correção e estado de tratamento, como aberto, em correção, resolvido ou risco aceito.

A plataforma consolida os achados em indicadores, incluindo sua distribuição por severidade e situação de tratamento. Como as respostas, evidências e decisões permanecem armazenadas, novas avaliações da mesma aplicação podem ser utilizadas para acompanhar alterações na postura de segurança e verificar se os problemas identificados foram corrigidos.

3. Aplicações Avaliadas

A SecureForge Web foi testada por meio da análise de diferentes aplicações web com o objetivo de demonstrar o fluxo de cadastro, análise, revisão e consolidação dos resultados, bem como examinar os indicadores de postura e os achados produzidos pela ferramenta.

Para cada aplicação-alvo, foram cadastrados o nome, a URL, o repositório (quando disponível) e a tecnologia. Em seguida executaram-se os módulos HTTP, Git e, opcionalmente, o de modelo de linguagem. Utilizou-se o perfil Essential v1.0 (Seção 2). Dois analistas coautores revisaram os 24 itens por alvo com acesso às sugestões; divergências foram resolvidas por consenso. Foram avaliadas 12 aplicações (labs vulneráveis, demos OSS e duas configurações da SecureForge Web), conforme a Tabela 2.

Dois analistas de segurança revisaram separadamente os 24 itens de cada alvo, após terem acesso às sugestões e evidências apresentadas pela plataforma. Cada item

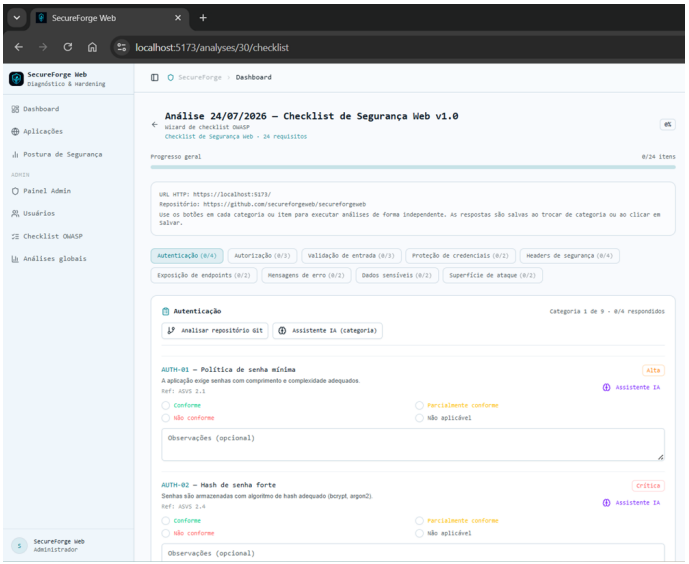

Figura 3. Wizard: sugestão, evidência e classificação do analista.

foi classificado como conforme, parcialmente conforme, não conforme ou não aplicável. As divergências foram discutidas até a obtenção de consenso, e a decisão consensual foi registrada como resultado final.

Ao todo, foram testadas 12 aplicações, organizadas em três grupos: intencionalmente vulneráveis, instâncias públicas de projetos de código aberto e duas configurações da própria SecureForge Web. A seleção buscou contemplar diferentes tecnologias e condições de análise, incluindo aplicações executadas em ambiente controlado, serviços acessíveis publicamente e uma comparação antes e depois de modificações de segurança.

Tabela 2. Aplicações Analisadas (Essential v1.0, 24/07/2026).

Aplicação	Tecnologia	Versão	Configuração
VAmPI	Python (Flask)	master (abr./2026)†	Local · lab vulnerável · API local
OWASP WebGoat (Java)	Java + Web	v2025.3	Local · lab vulnerável · app local
OWASP NodeGoat	Node.js + MongoDB	v1.4	Local · lab vulnerável · app local
OWASP WebGoat (PHP)	PHP + Web	release lab†	Local · lab vulnerável · app local
DVWA	PHP + JS	2.5	Local · security=*low* · app local
OWASP Mutillidae II	PHP + Web	v2.7.11	Local · lab vulnerável · app local
OWASP Juice Shop	Node.js + Angular/TS	v20.1.1	Local · lab vulnerável · app local
Ghost CMS	JS/TS	instância pública*	Pública · demo · demo.ghost.io
Gitea	Go + TS/JS	instância pública*	Pública · demo · demo.gitea.com
Mattermost	TS/JS + Go	instância pública*	Pública · comunidade · community.mattermost.com
SecureForge Web	TS/JS	main (23–24/07)‡	Local · baseline · localhost:5173
SecureForge Web v2	TS/JS	main (24/07)‡	Local · pós-hardening parcial · localhost:5173

4. Avaliação

A avaliação foi realizada em quatro etapas. Na primeira, foram identificados a aplicação-alvo e, quando disponível, seu repositório público. Para cada alvo, foram cadastrados na SecureForge Web o nome da aplicação, sua URL, o endereço do repositório e a tecnologia

predominante. Na segunda etapa, foi selecionado o perfil *Checklist Essential SecureForge v1.0*, composto por 24 itens distribuídos em 9 categorias. Cada item admite quatro classificações: conforme, parcialmente conforme, não conforme e não aplicável.

Na terceira etapa, foram executados os módulos de análise HTTP e de repositório Git. O primeiro examinou informações observáveis na resposta da aplicação, enquanto o segundo analisou padrões presentes no código-fonte disponível. Em todas as avaliações, também foi utilizado o modelo `gpt-4o-mini`, mantendo-se os mesmos parâmetros de configuração. O modelo recebeu o item avaliado e as evidências disponíveis e apresentou uma classificação sugerida. Essas sugestões não foram registradas automaticamente como decisões finais.

Na quarta etapa, dois analistas de segurança revisaram separadamente os 24 itens de cada alvo, após terem acesso às sugestões e evidências apresentadas pela plataforma. Após a conclusão das avaliações individuais, as classificações divergentes foram discutidas até a obtenção de consenso. A classificação consensual foi registrada na SecureForge Web e utilizada para calcular o score de postura e gerar os achados correspondentes.

4.1. Resultados

A Tabela 3 resume os indicadores dos 12 alvos. O score é $(\#\text{conforme} + \#\text{N/A})/24$; N/A no numerador pode inflar o valor e não foi contado por alvo (leitura operacional - Seção 4.2). Itens parcial/não conformes geram achados; C/A/M/B = crítica/alta/média/baixa.

Tabela 3. Indicadores de postura obtidos para os alvos avaliados.

Aplicação	Perfil	Score	Achados	C/A/M/B
VAmPI	Lab. vulnerável	25%	18	4 / 9 / 5 / 0
OWASP WebGoat (Java)	Lab. vulnerável	29%	17	5 / 8 / 4 / 0
OWASP NodeGoat	Lab. vulnerável	33%	16	4 / 8 / 3 / 1
OWASP WebGoat (PHP)	Lab. vulnerável	33%	16	5 / 6 / 5 / 0
DVWA	Lab. vulnerável	38%	15	4 / 7 / 4 / 0
OWASP Mutillidae II	Lab. vulnerável	38%	15	5 / 6 / 4 / 0
Ghost CMS	Demo OSS	42%	14	3 / 7 / 4 / 0
Gitea	Demo OSS	42%	14	3 / 7 / 3 / 1
OWASP Juice Shop	Lab. vulnerável	46%	13	4 / 6 / 3 / 0
Mattermost	Demo OSS	58%	10	3 / 5 / 1 / 1
SecureForge Web	Autoavaliação	63%	9	1 / 4 / 4 / 0
SecureForge Web v2	Autoavaliação	**75%**	6	1 / 2 / 3 / 0

Os laboratórios intencionalmente vulneráveis apresentaram scores entre 25% e 46%, com 13 a 18 achados. O VAmPI apresentou o menor score, 25%, e o maior número de achados, 18. WebGoat, NodeGoat, DVWA e Mutillidae II ficaram entre 29% e 38%, enquanto o Juice Shop apresentou score de 46% e 13 achados. Em todos esses alvos, houve predominância de achados de severidade crítica ou alta.

Entre as instâncias públicas de projetos de código aberto, Ghost CMS e Gitea obtiveram score de 42% e 14 achados cada, enquanto o Mattermost apresentou 58% e

10 achados. Esses resultados refletem as classificações consensuais dos analistas após a revisão das sugestões e evidências produzidas pela plataforma.

Na autoavaliação da SecureForge Web, o score aumentou de 63% na configuração inicial para 75% após o hardening parcial, enquanto o número de achados diminuiu de nove para seis. Esse resultado demonstra o uso da plataforma em um ciclo de avaliação, modificação e reavaliação da mesma aplicação.

4.2. Limitações e Ameaças à Validade

Os scores da Tabela 3 representam as classificações dos 24 itens do Essential v1.0 segundo a regra da plataforma (conforme+N/A), e não uma medida completa da segurança dos alvos nem a eficácia isolada dos módulos. Tratar N/A como equivalente a conforme é escolha de produto e pode inflar o indicador; um score só-aplicáveis e a coluna N/A por alvo exigem nova coleta.

Quanto à validade interna, os analistas (coautores) viram sugestões e evidências antes de classificar: o estudo avalia o fluxo assistido + revisão humana, mas não isola a acurácia de HTTP, Git ou IA, nem registra concordância formal entre avaliadores. Quanto à validade externa, usaram-se 12 alvos, um perfil de checklist e um modelo de linguagem; instâncias públicas e o ciclo de hardening apenas na própria SecureForge Web limitam a generalização.

5. Conclusão

A SecureForge Web materializa um fluxo operacional de checklist OWASP guiado, automação assistida com três escopos complementares (HTTP, Git e IA), evidências estruturadas por item e consolidação em achados, postura e PDF. A avaliação experimental sobre doze aplicações mostrou scores entre 25% e 46% nos laboratórios vulneráveis, entre 42% e 58% nas demos de software livre, e de 63% para 75% na autoavaliação da própria plataforma após hardening parcial. Os resultados confirmaram operação ponta a ponta e utilidade para equipes sem processo formal de AppSec.

Como trabalhos futuros, além da expansão das heurísticas Git e da integração com varredura de dependências, planeja-se: (i) estudo com referência independente (*ground truth* ou baseline manual sem sugestões), medindo acurácia, FP/FN, taxa de aceitação e tempo de revisão; (ii) reportar N/A por alvo e um score só-aplicáveis; (iii) avaliação com analistas externos (não coautores); (iv) imagem de VM com a ferramenta e um alvo de exemplo pré-carregados.

Agradecimentos

Esta pesquisa foi parcialmente financiada, conforme previsto nos Arts. 21 e 22 do decreto no. 10.521/2020, nos termos da Lei Federal no. 8.387/1991, através do convênio no. 015/2025, firmado entre ICOMP/UFAM, Digiboard Eletrônica da Amazônia Ltda e Lenovo Tecnologia Brasil Ltda. O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001 e parcialmente financiado pela Fundação de Amparo à Pesquisa do Estado do Amazonas – FAPEAM – por meio do projeto POSGRAD 2025/2026.

Referências

Akrout, R., Alata, E., Kaaniche, M., and Nicomette, V. (2014). An automated black box approach for web vulnerability identification and attack scenario generation. *Journal of the Brazilian Computer Society*, 20:artigo 4.

OWASP (2021). Web security testing guide.

OWASP (2025). Application security verification standard (asvs). Versão 5.0.0. Acessado em: 12 de agosto de 2026.

Rennhard, M., Kushnir, M., Favre, O., Esposito, D., and Zahnd, V. (2022). Automating the detection of access control vulnerabilities in web applications. *SN Computer Science*, 3:artigo 376.

Wen, S.-F. and Katt, B. (2023). A quantitative security evaluation and analysis model for web applications based on owasp application security verification standard. *Computers & Security*, 135:artigo 103532.

Zech, P., Felderer, M., and Breu, R. (2019). Knowledge-based security testing of web applications by logic programming. *International Journal on Software Tools and Technology Transfer*, 21(2):221–246.

Apêndice A. Demonstração da Ferramenta

A demonstração no Salão de Ferramentas (modalidade Código Aberto) cobre o fluxo ponta a ponta: cadastro do alvo, checklist Essential v1.0, módulos HTTP/Git/IA (opcional), revisão humana, achados e indicadores de postura (com PDF se o tempo permitir), usando um alvo pré-configurado. Equipamento: notebook com a ferramenta em execução (Node.js 22, pnpm, PostgreSQL 16), projetor ou cabo HDMI/USB-C; Internet desejável, com contingência local. O código-fonte e as instruções de instalação estão em repositório público[2], e o vídeo de instalação e funcionamento está disponível publicamente[3].

Apêndice B. Declaração de Uso de IA Generativa

Em conformidade com o Código de Conduta para Autores em Publicações da SBC, declaramos uso de modelos de linguagem: (i) como componente da SecureForge Web (sugestão de classificações no checklist); e (ii) como apoio à revisão linguística do manuscrito. Os autores assumem total responsabilidade pelo conteúdo final.

[2] `https://github.com/secureforgeweb/secureforgeweb`.
[3] `https://drive.google.com/drive/folders/1oJRC9_3Zjx5ahBdgdXSajCKXhYjytKWX`.